Южная математическая смена 2018 в «Сириусе»

Курс «Машинное обучение»

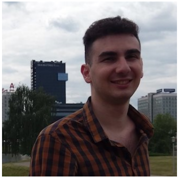
Лекция первая

«Введение в машинное обучение» [1]

14 ноября, 2018

[1] Автор: Мурат Апишев (mel-lain@yandex.ru)

Ваши преподаватели



- **Мурат Апишев**
- ВМК МГУ, ФКН ВШЭ, ФУПМ МФТИ, ШАД
- Яндекс.Дзен, ЛМИ МФТИ, Яндекс.Поиск

- **Надежда Зуева**
- ФИВТ МФТИ, ФПМИ МФТИ
- iPavlov, ABBYY

- **Татьяна Гайнцева**
- ФИВТ МФТИ, ШАД, Сбербанк
- LAMBDA ВШЭ

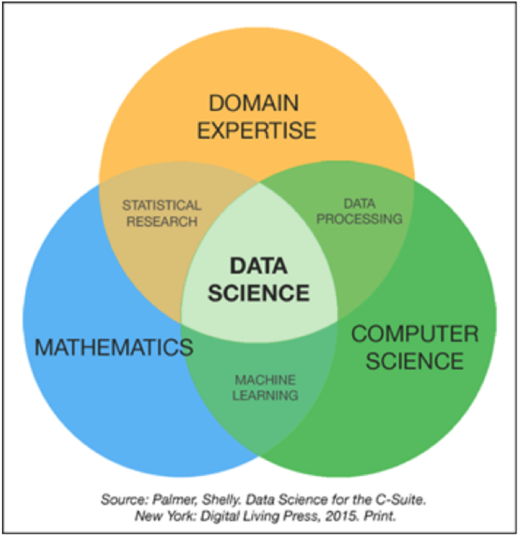
Что такое машинное обучение

Машинное обучение (machine learning, ML) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач

Что это означает:

- Есть некоторая задача, например
  - поиск текстов
  - распознавание изображений
  - прогноз цен на акции
  - написание стихов
- Человек умеет решать задачу хорошо, но делает это очень медленно
- Хочется на примере человека научить компьютер решать эти задачи намного быстрее с небольшой потерей качества

Машинное обучение и анализ данных

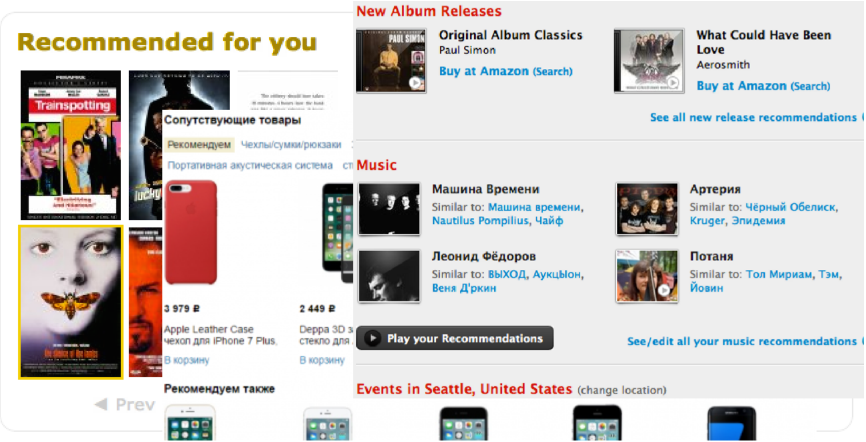


Пример: поисковые системы

- Поисковые системы прочно вошли в нашу жизнь, мы пользуемся ими каждый день
- Раньше поиск был ограничен текстами
- В последние годы развитие нейронные сетей позволило создать поиск по изображениям
- Поиск – сложная система, в которой решается несколько больших задач машинного обучения

Ссылка на источник картинки

Пример: рекомендации



Пример: машинный перевод

- Учить компьютер переводить тексты вместо человека пробовали ещё с середины прошлого века
- До недавнего времени результаты были не слишком хороши, даже переводчики Google и Yandex делали некачественные переводы
- Машинный перевод оказался одной из областей анализа текстов, в которых развитие нейронных сетей привело к качественному прорыву

Ссылка на источник картинки

Пример: прогнозирование

Прогнозировать можно пытаться буквально всё, что угодно:

- Выдавать кредит или нет?
- Отзыв на фильм хороший или нет?
- Чем может быть болен этот человек?
- Как разложить эти песни по жанрам?
- Какая модель автомобиля изображена на фотографии?
- Как будет меняться цена портфеля акций на бирже?
- Что сейчас произнёс этот человек?

В широком смысле прогнозирование включает в себя и поисковые системы, и рекомендации

Исторические вехи машинного обучения

- Математический аппарат, который лёг в основу машинного обучения, стал появляться ещё в 18-м веке
- Искусственный интеллект и распознавание образов образов появились вместе с первыми компютерами
- **1943** – модель искусственного нейрона МакКаллока-Питтса
- **1960** – первый нейрокомпьютер (двуслойная сеть)
- **1970**-е – генетические алгоритмы
- **1977** – восстановление смеси распределений, EM-алгоритм
- **1963, 1992** – метод опорных векторов (SVM)

Современное положение дел

- **1982** – рекурентные нейронные сети (RNN)
- **1998** – свёрточные нейронные сети (CNN)
- **1999** – градиентный бустинг
- **2001** – случайный лес
- **1986** – метод обратного распространения ошибок (BackProp)
- **1965-2007** – глубокие нейронные сети
- **Последние 10 лет: бум нейронных сетей**
  - Обучение нейронных сетей перешло с процессоров на видеокарты
  - Сбор и использование больших массивов данных (Big Data)
  - Появилось большое количество разнообразных архитектур (Resnet, GAN, Transformer, VAE)
  - Вместе с нейросетями активно развивается *обучение с подкреплением* – основная надежда робототехники

Что требуется от хорошего аналитика

1. Владением математическим аппаратом:
   - линейная алгебра и аналитическая геометрия
   - математический анализ
   - теория вероятности и математическая статистика
   - методы оптимизации
   - алгоритмы и структуры данных
2. Знание предметной области, например:
   - анализ данных в медицине/фармацевтике требует некоторого владения химией и биологией
   - для анализа текстов (NLP) нужно разобраться в грамматике и лингвистике
3. Умение программировать:
   Python/Matlab/R/Julia/C++/Java/C#/Scala/Go

Математические основы

- Указанные выше области математики читаются в университетах в течение нескольких лет
- Тем не менее, для практического использования ML обычно хватает самых базовых понятий и алгоритмов
- К таким относятся:
  - вектор и матрица
  - матричные операции
  - определение и смысл производной
  - градиент и градиентный спуск
  - вероятность и распределение вероятности
  - выборка и её статистики
- Как только разберёмся с ними, попробуем формально поставить задачи машинного обучения

Вектор

- Вектор – это величина, характеризующаяся величиной и направлением
- Вектор – это направленный отрезок
- Вектор – это элемент векторного (линейного) пространства
- Вектор – это массив из n элементов (в случае конечномерного пространства)

Все эти определения верны и обозначают одно и то же!

Проще всего представить вектор a как массив вещественных чисел длины n, над которым определены операции

- по-элементного сложения: $a + b = c, \quad c_i = a_i + b_i, \ i = \overline{1, n}$
- умножения на число (скаляр): $a * k = d, \quad d_i = a_i * k, \ i = \overline{1, n}, \ k \in \mathbb{R}$

Ссылка на источник картинки

Связь с геометрией

- Вектор в геометрии – направленный отрезок
- Без ограничения общности можно считать, что он исходит из точки начала координат
- Тогда, в случае плоскости, он определяется двумя числами-координатами, в случае трёхмерного пространства – тремя
- Сложение векторов в геометрии идентично описанной выше операции по-элементного сложения

Скалярное произведение

- *Скалярное произведение векторов* – ещё одно фундаментальное понятие линейной алгебры
- Обычно определяется в школьной геометрии следующим образом

$$\langle a, b\rangle = |a| \cdot |b| \cdot \cos \angle(a, b)$$

- $|\cdot|$ – это *норма вектора*, можно понимать её как длину вектора. Нормы бывают разные, наиболее известная – евклидова

$$|a| = \sqrt{\sum_{i=1}^{n} a_i^2}$$

- В алгебраической форме СП выглядит так

$$\langle a, b\rangle = \sum_{i=1}^{n} a_i \cdot b_i$$

Метрика

- *Метрика* – функция $d(x, y)$, удовлетворяющая следующим аксиомам:
  1. $d(x, y) = 0 \Leftrightarrow x = y$
  2. $d(x, y) == d(y, x)$
  3. $d(x, z) \leqslant d(x, y) + d(y, z)$ – *неравенство треугольника*
- Метрика – это расстояние между двумя объектами: числами, векторами, матрицами
- Примеры:
  - $d(x, y) = 1$ при $x = y$, иначе – 0
  - $d(x, y) = |x - y|$, $x, y \in \mathbb{R}$
  - $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|, x, y \in \mathbb{R}^n$
  - $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, x, y \in \mathbb{R}^n$ – *евклидово расстояние*

Матрица

Матрица (над полем $\mathbb{R}$) -— математический объект, записываемый в виде прямоугольной таблицы элементов из $\mathbb{R}$, которая представляет собой совокупность строк и столбцов, на пересечении которых находятся эти элементы

Вектор = одномерная матрица! Но важно, столбец или строка

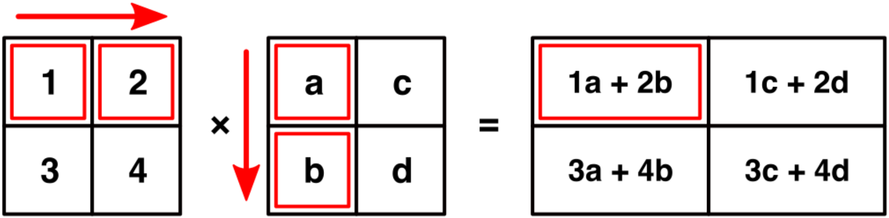
Ссылка на источник картинки

Основные матричные операции

- **Транспонирование**: A^T – матрица «проворачивается» вокруг главной диагонали на 180 (строки становятся столбцами)
- **Обращение**: A^{-1} – матрица, умножение на которую матрицы A даст единичную матрицу (определена только для подкласса квадратных матриц)
- **Умножение на число** – аналогично векторам
- **Матричное умножение** – если есть две матрицы $A \in \mathbb{R}^{n \times m}$ и $B \in \mathbb{R}^{m \times \ell}$, то их произведение будет матрицей $C \in \mathbb{R}^{n \times \ell}$, элементы которой определяются выражением

$$c_{ij} = \sum_{k=1}^{m} a_{ik} \cdot b_{kj}$$

Произведение матриц



Перемножаться могут любые матрицы, у которых совпадает *промежуточная размерность* – число столбцов у первой и число строк у второй

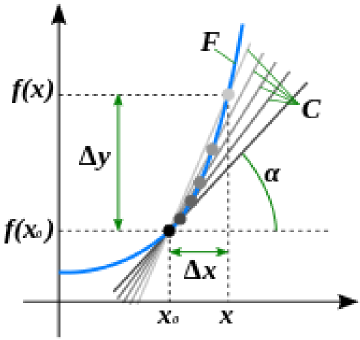
Ссылка на источник картинки

Производная функции

- Пусть функция f определена и гладка в некоторой окрестности точки x
- Тогда производной $f'(x)$ функции f в точке x называется величина

$$f'(x) = \frac{\Delta f}{\Delta x}, \quad \Delta x \to 0$$

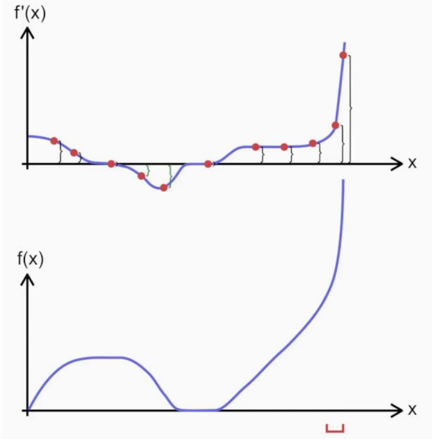
- Здесь Δx означает длину очень маленького шага от точки x
- Δy показывает, насколько сильно изменилось значение функции f при таком шаге аргумента



Смысл производной

- Производная характеризует скорость роста функции в точке:
  - $f'(x) = 0$ – функция в точке идёт параллельно оси аргумента
  - $f'(x) > 0$ – функция растёт, чем выше значение, тем быстрее
  - $f'(x) < 0$ – функция убывает, чем выше модуль значения, тем быстрее
- В уравнении прямой $y = k \cdot x + b$ величина k называется *угловым коэффициентом*
- С геометрической точки зрения, производная в точке x – угловой коэффициент касательной к графику f в x
- Изменение знака производной означает изменение характера монотонности функции (до изменения возрастала, после – стала убывать)

Взаимоотношении функции и её производной



Нули производной соответствуют точкам, подозрительным на экстремум.

Правила дифференцирования

Дифференцирование функции – процесс вычисления производной. Из определения производной в выведены следующие правила дифференцирования:

- $f(x) = C \Rightarrow f'(x) = 0$
- $f(x) = x^\alpha \Rightarrow f'(x) = \alpha x^{\alpha-1}$
- $f(x) = a^x \Rightarrow f'(x) = a^x \ln a$
- $f(x) = \log_a x \Rightarrow f'(x) = \dfrac{1}{x \ln a}$
- $f(x) = e^x \Rightarrow f'(x) = e^x$
- $f(x) = \operatorname{tg} x \Rightarrow f'(x) = \dfrac{1}{\cos^2 x}$

- $f(x) = \sin x \Rightarrow f'(x) = \cos x$
- $f(x) = \cos x \Rightarrow f'(x) = -sinx$

- $(f+g)' = f' + g'$
- $(Cf)' = Cf'$
- $(fg)' = f'g + fg'$
- $\left(\dfrac{f}{g}\right)' = \dfrac{f'g - fg'}{g^2}, \quad g \neq 0$
- $(f(g(x)))' = f'(g(x))g'(x)$

Частная производная

- *Частная производная* – обобщение понятия производной на функции многих переменных
- Берётся по одной из переменных, равна отношению прироста функции к бесконечно малому приросту этой переменной
- Допустим, функция – скорость автомобиля s, которая зависит от объёма двигателя c и массы корпуса m
- Зафиксируем объём двигателя и увеличим массу корпуса
- Измерим, насколько поменялась скорость при таком изменении
- Посчитаем отношение приростов, устремив изменение массы к 0
- Получим частную производную скорости по массе корпуса:

$$\frac{\partial s}{\partial m} = \lim_{\Delta m \to 0} \frac{s(c, m + \Delta m) - s(c, m)}{\Delta m}$$

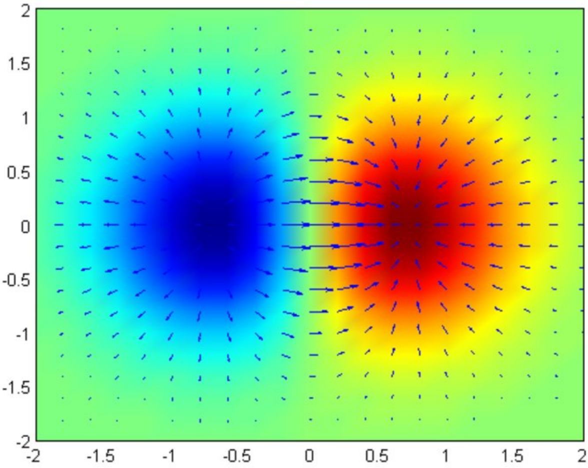
Градиент функции

- *Градиентом* функции, зависящей от n переменных, называется вектор её частных производных по этим переменным:

$$f(x, y, z) \quad \Rightarrow \quad \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

- Вектор градиента в каждой точке показывает направление наибольшего роста функции
- Точка является *локальным максимумом* функции, если существует такая её окрестность, где функция везде меньше, чем в этой точке
- *Локальный минимум* определяется аналогично
- Точки максимума/минимума называются *экстремумами*
- Градиент позволяет искать локальные экстремумы функции

Визуализация градиента 2D функции



Оптимизация функции

- Поиск экстремумов функции называется её *оптимизацией*. Обычно ищут минимум:

$$f(x) \to \min_x$$

- x может быть числом, может быть вектором или матрицей
- Для максимизации достаточно найти минимум $-f(x)$
- В начале поиска фиксируется некоторая точка x_0
- Дальше из этой точки нужно выбрать следующую точку, которая будет ближе к точке экстремума, которую мы ищем
- Существуют различные стратегии того, как делать эти шаги
- Разные алгоритмы имеют разные оценки скорости достижения какого-нибудь локального экстремума с точки зрения числа шагов и сложности их вычисления

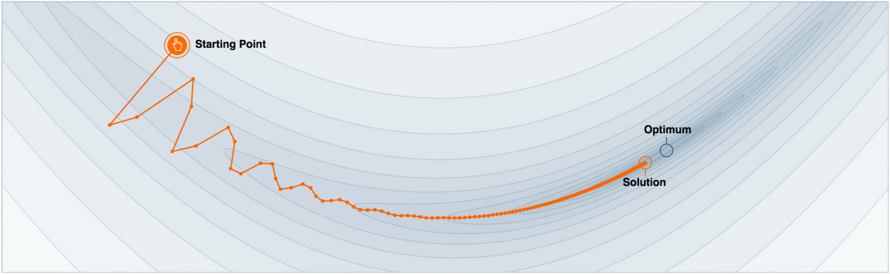
Градиентный спуск

- *Метод градиентного спуска* – один из наиболее простых алгоритмов поиска локального экстремума заданной дифференцируемой функции f
- **Идея:** будем двигаться в направлении наибольшего убывания функции, которое задаётся антиградиентом $-\nabla f$
- На каждом шаге будем искать новую точку по предыдущей исходя из выражения

$$x^{i+1} = x^i - \lambda^i \nabla f(x^i)$$

- Выбор λ^i сильно влияет на качество и время работы алгоритма
- Слишком маленькое – будем двигаться медленно, слишком большое – можем «проскочить» узкий минимум
- Разные стратегии выбора λ^i определяют разные методы оптимизации

Градиентный спуск



На каждом шаге смещаемся в точку, которая сейчас кажется оптимальной

Ссылка на источник картинки

Вероятность

- *Вероятность* – численная мера возможности наступления некоторого события
- Существуют различные определения вероятности
- Рассмотрим частотное определение: вероятность наступления события A из некоторого (потенциально бесконечного) множества событий определяется как:

  $$p(A) = \lim_{n\to\infty} \frac{n_a}{n},$$

  где n – общее число испытаний, n_a – число раз, когда в испытании наблюдалось событие A
- Самый простой пример: подбрасывание монетки (два возможных события)

Вероятность

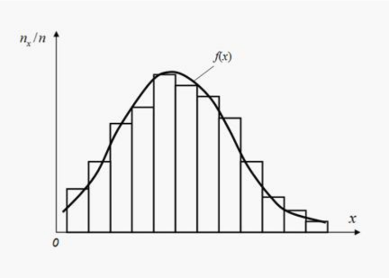
- Вероятность изменяется от 0 до 1 включительно
- Если у нас есть совокупность всевозможных событий X, то $p(X) = 1$
- Вероятность невозможного события $p(\varnothing) = 0$
- *Случайная величина* – величина, значения которой представляют собой исходы случайного события
- Исходы бросков монеты – это случайная величина
- Эта величина – *дискретная*, потому что множество исходов состоит из отдельных объектов (орёл/решка)
- Случайная величина может быть *непрерывной* – например, рост случайного человека (ось роста)

Вероятностное распределение

- Очевидно, что среди всех возможных ростов людей какие-то встречаются часто, какие-то – реже
- *Вероятностное распределение* – это закон, описывающий область значений случайной величины и вероятности их появления
- Распределение можно задавать через *функцию распределения* и
  - *закон распределения* – для дискретного распределения
  - *плотность* – для непрерывного
- Разберёмся сперва с дискретной. Пусть мы замеряем число людей с ростом 140 см, 150 см, . . . , 220 см
- Пусть по медицинской статистике получается на 1000 человек:
  1. 140 см: 0.03 от общего числа
  2. 150 см: 0.07 от общего числа
  3. . . .
  4. 220 см: 0.02 от общего числа
- **Это и есть закон распределения – у какого исхода какая вероятность**

Плотность распределения

- Закон распределения можно изобразить гистограммой (прямоугольники)
- По оси Ox отложены возможные исходы (в нашем случае рост), по Oy – вероятность
- А теперь представим, что мы дробим Ox не по 10 см, а на очень малые промежутки
- Наконец, устремим длину промежутков к нулю, и получим кривую, как на рисунке
- Это будет плотность нашего распределения – функция, которая в каждой точке равна вероятности
- Это вероятность того, что случайная величина в очередном эксперименте примет значение, равное этой точке



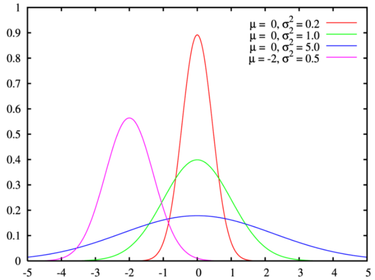
Ссылка на источник картинки

Нормальное распределение

- *Нормальное (гауссовское) распределение* – самое важное и часто используемое распределение в любой прикладной области

У него есть два *параметра* – *математическое ожидание* μ и *среднеквадратичное отклонение* σ, функция плотности имеет вид

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



μ говорит о том, где у графика плотности «середина»

А σ – насколько далеко можно уйти от μ

- Проще всего представить, что мы хотим генерировать случайные числа (*сэмплирование*)
- Тогда график плотности определяет то, в каких диапазонах и с какой вероятностью мы будем получать числа

Ссылка на источник картинки

Задача обучения по прецедентам

- *Обучение по прецедентам* = *обучение с учителем*
- Есть множество объектов X и множество ответов Y
- Также есть неизвестная функция y, которая каждому элементу X ставит в соответствие какой-то элемент y
- **Пример:** X – фотографии автомобилей, Y – марки автомобилей, y – соответствие между фотографией авто и его маркой
- В задаче обучения с учителем имеем *обучающую выборку* $x_1, \dots, x_\ell \in X$, для каждого объекта которой известен правильный ответ $y_i = y(x_i)$
- Цель: построить на основе обучающей выборки алгоритм $a(x)$, который будет как можно лучше приближать функцию y на всём множестве X

Ссылка на источник материала

Признаковое описание объекта

- **Вопрос:** а что такое объект в X?
- Каждый объект характеризуется своим *признаковым описанием*
- *Признак* – это некоторая функция $f_j(x)$, которая принимает на данном объекте какое-то значение
- **Пример:** объект – человек, признаки – температура, рост, пол, вес, цвет волос и т.д.
- Признаковое описание – это совокупность признаков, которые мы используем в данной задаче (для медицинской диагностики вес важен, а литературные предпочтения – не очень)
- В зависимости от своей области значений, признаки бывают
  1. бинарными (пол: мужской/женский)
  2. категориальные (цвет волос: чёрный/каштановый/красный)
  3. порядковыми (оценка за экзамен: 1/2/3/4/5)
  4. количественными (вес: R)

Признаковое описание объекта

- Для решения задач машинного обучения объект реального мира представляется в виде значений набора признаков $f_1, \ldots, f_n$
- Тогда обучающая выборка $\{x_i\}_{i=1}^{\ell}$ описывается матрицей F

$$F = \begin{bmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \vdots & \vdots & \vdots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{bmatrix}$$

- Эта матрица состоит только из чисел, если есть категориальные признаки, их надо превратить в числовые
- Самый простой вариант – заменить на несколько бинарных:
  - **было:** один признак «цвет волос» со значениями «чёрный», «каштановый», «красный»
  - **стало:** три признака «цвет волос чёрный», «цвет волос каштановый», «цвет волос красный» со значениями 0 и 1

Типы задач

В зависимости от типа множества ответов Y выделяют различные типы задач обучения с учителем:

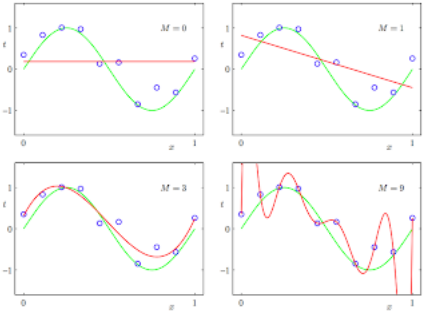
- Задачи классификации:
  - $Y = \{0, 1\}$ – бинарная (товар хороший или плохой?)
  - $Y = \{1, \dots, M$ – многоклассовая без пересечений классов (какая марка у этого автомобиля?)
  - $Y = \{0, 1\}^M$ – многоклассовая с пересечением – каждый объект может относиться к нескольким классам (какие жанры у этого фильма?)
- Задачи регрессии:
  - $Y = \mathbb{R}$ (какой рост у этого человека?)
- Задачи ранжирования:
  - Y – конечное упорядоченное множество (как эти веб-страницы упорядочены по релевантности этому поисковому запросу?)

Этапы решения задачи обучения с учителем

- Поделим нашу выборку на две части: на *обучающую* и *тестовую*
- Первая нужная для того, чтобы на ней обучить модель
- Вторая – чтобы проверить качество работы этой модели
- Проверять качество работы на обучающей выборке нельзя – можно получить *переобучение*
- Это ситуация, когда модель слишком сильно подстроилась под данные обучения и не умеет никак свои знания обобщать
- Рассмотрим на примере задачи регрессии – пытаемся предсказывать значения неизвестной функции одного аргумента по нескольким точкам

Переобучение и недообучение

- Наша модель – функция-многочлен
- У модели есть *параметры* – то, что обучается, в нашем случае, это коэффициенты многочлена
- Ещё у модели есть *гиперпараметры* – то, что мы выбираем заранее, здесь это степень многочлена
- Попробуем обучать модели с разными степенями (как – узнаем потом)
- Линейный многочлен слишком простой
- Многочлен большой степени подстраивается под точки обучающей выборки и совсем не похож на реальную функцию



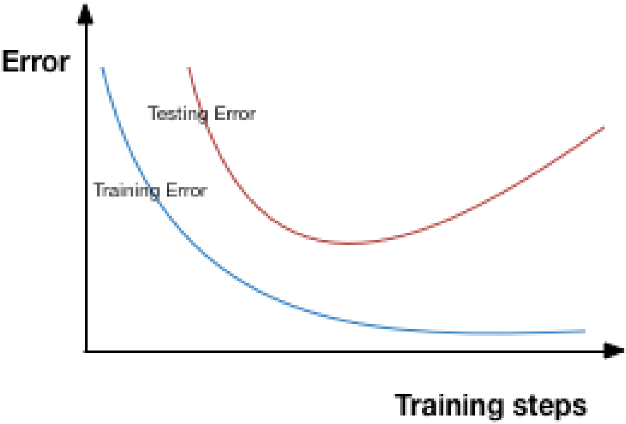
Как контролировать качество

- Обучая модель, мы стараемся заставить её как можно качественнее решать задачу на тренировочной выборке
- Для этого формулируется *функционал качества*

$$L = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathrm{err}(x_i)$$

- Он обозначает насколько сильно текущая модель ошибается на всех обучающих объектах в среднем
- Для каждой задачи функция err определяется по-своему
- Например, для бинарной классификации это может быть просто 0, если реальный и предсказанный классы объекта совпали, и 1 – иначе
- Тогда L будет иметь смысл точности – того, насколько большую долю объектов модель классифицирует неправильно
- А если посчитаем L на тестовой выборке, то узнаем, насколько хорошо модель классифицирует новые данные!

Вот так выглядит переобучение

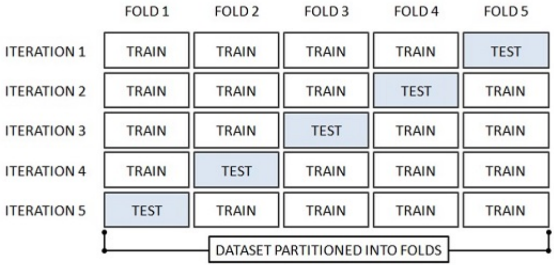


Ссылка на источник картинки

Кросс-валидация

- Выбирая в качестве тестовых данных разные объекты выборки, мы можем получить сильно разные значения тестового качества
- Поэтому при подборе параметров модели используется *кросс-валидация*
- Самый распространённый вариант такой: все данные случайно разбиваются на несколько частей, дальше каждая часть по-очереди используется в качестве тестовой, а все остальные – в качестве обучения
- На каждой такой итерации будут получаться свои наилучшие параметры, которые нужно усреднить
- Средняя ошибка L, посчитанная на кросс-валидации, более устойчива, чем на одном разбиении

Кросс-валидация

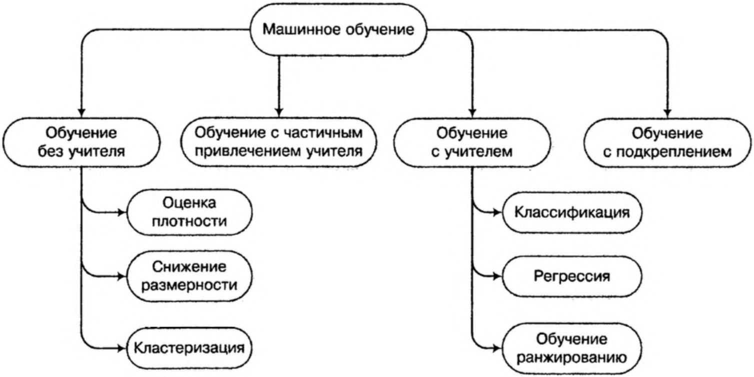


- Есть проблема, что мы в процессе обучения подстраиваемся под тест
- Это тоже переобучение
- Поэтому часто данные делят на три части: обучение, тест и валидация
- Все описанные операции производятся над валидационными подвыборками
- Качестве на тестовой выборке измеряется в самом конце обучения

Ссылка на источник картинки

Какие ещё бывают типы задач

Задачи машинного обучения не ограничиваются обучением с учителем:



Ссылка на источник картинки

Итоги занятия

- Задачи машинного обучения решаются повсюду в современном мире, часто это незаметно для пользователей
- Для того, чтобы заниматься ML, нужно знать математику и уметь программировать (хоть чуть-чуть)
- Набор базовых знаний достаточно небольшой, мы сегодня с ним ознакомились и будем закреплять по ходу дальнейшего изучения
- Одна из основных задач ML – обучение с учителем, где мы по размеченным данным пытаемся предсказывать ответы на новых
- Для решения задач ML существует много различных подходов и моделей, о которых мы поговорим в следующих лекциях

Успехов!:)